WHAT IS CLAIMED IS:

1.    A method for facilitating prevention of interception of incoming data that is provided for a software application, comprising the steps of:

        providing insertion data for insertion as part of the incoming data;

        storing the generated insertion data; and

        filtering received incoming data containing actual data and the insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data.

2.    The method of claim 1 including the step of processing the actual data resultant from filtering for use by the software application.

3.    The method of claim 1 including the steps of:

        receiving the generated insertion data and actual data from a data input source; and

        queuing the insertion data with the actual data for output as the incoming data.

4.    The method of claim 1 including the step of analyzing foreground indication data and enabling generation of the insertion data in response to the foreground indication data.

5.    The method of claim 1 including the step of controlling timing of insertion data generation and output based on data queue parameters.

6.    The method of claim 1 wherein the step of storing includes storing the generated insertion data as entries in a first in first out (FIFO) buffer and wherein the step of filtering further includes comparing the incoming data to at least one

11

entry of stored insertion data and deleting the at least one entry of stored data and deleting the incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

7.      The method of claim 1 including the steps of:
        storing a list of data representing data to be used as randomization data;
        randomly selecting the randomized data from the list of data; and
        formatting the randomized data as insertion data in a same format as actual data.

8.      The method of claim 1 wherein the actual data is keystroke data from a keystroke source, wherein the random insertion is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

9.      The method of claim 1 wherein the step of providing includes:
        providing the insertion data, under control of the software
        application that is to receive the incoming data.

10. A method for facilitating prevention of interception of incoming data that is provided for a software application, comprising the steps of:

analyzing foreground indication data and enabling generation of the insertion data in response to the foreground indication data;

storing a list of data representing data to be randomized;

selecting data from the list of data as random insertion data;

providing selected insertion data for insertion as part of the incoming data;

formatting the insertion data in a same format as actual data;

storing the generated insertion data;

mixing the insertion data with the incoming data; and

filtering received incoming data containing actual data and the random insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data.

11. The method of claim 10 including the step of processing the actual data resultant from filtering for use by the software application.

12. The method of claim 10 including the steps of:

receiving the generated random insertion data and actual data from a data input source; and

queuing the random insertion data with the actual data for output as the incoming data.

13. The method of claim 10 including the step of controlling timing of random insertion data generation and output based on data queue parameters.

14. The method of claim 10 wherein the step of storing includes storing the generated insertion data as entries in a first in first out (FIFO) buffer and wherein the step of filtering further includes comparing the incoming data to at least one
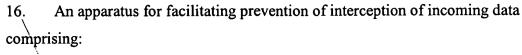
entry of stored insertion data and deleting the at least one entry of stored random data and deleting the incoming data when the stored data is the same as the incoming data, and outputting the incoming data to a data processor when the stored data from the entry is different from the incoming data.

15. The method of claim 14 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

16. An apparatus for facilitating prevention of interception of incoming data comprising:

means for providing insertion data for insertion as part of the incoming data;

means for storing the provided insertion data; and

means for filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data for use by a software application.

17. The apparatus of claim 16 including means, operatively coupled to the means for filtering, for processing the actual data resultant from filtering for use by the software application.

18. The apparatus of claim 16 wherein the means for filtering receives the provided insertion data and actual data, and wherein the apparatus includes means for queuing the insertion data and the actual data for output as the incoming data.

19. The apparatus of claim 17 wherein the means for providing insertion data generates the insertion data in response to foreground indication data.

20. The apparatus of claim 18 wherein the means for providing the insertion data includes timing means for controlling timing of random insertion data generation and output based on data queue parameters.

21. The apparatus of claim 16 wherein the means for storing includes a first in first out (FIFO) buffer that stores the generated random insertion data as entries therein and wherein the means for filtering incoming data further includes means for comparing the incoming data to at least one entry of stored random insertion data and deleting the at least one entry of stored random data and deleting the

incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

22.   The apparatus of claim 17 including:

        means for storing a list of data representing data to be randomized; and

        means, operatively coupled to the means for storing the list of data, for selecting data from the list of data as random insertion data; and

        means for formatting the random insertion data in a same format as actual data.

23.   The apparatus of claim 22 wherein the actual data is keystroke data from a keystroke source, wherein the random insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

24. An apparatus for facilitating prevention of interception of incoming data comprising:

means for storing a list of data representing data to be randomized;

means, operatively coupled to the means for storing the list of data, for randomly selecting data from the list of data

means for providing random insertion data, using the randomly selected data from the list of data, for insertion as part of the incoming data in response to foreground indication data;

means for formatting the random insertion data in a same format as actual data;

means for mixing the insertion data with the actual data;

means for storing the generated random insertion data; and

means for filtering received incoming data containing actual data and the random insertion data by comparing stored generated random insertion data with incoming data to determine which data is actual data for use by a software application.

25. The apparatus of claim 24 including means, operatively coupled to the means for filtering, for processing the actual data resultant from filtering for use by the software application.

26. The apparatus of claim 24 wherein the means for filtering receives the generated random insertion data and actual data, and wherein the apparatus includes means for queuing the random insertion data and the actual data for output as the incoming data.

27. The apparatus of claim 24 wherein the means for generating the random insertion data includes timing means for controlling timing of random insertion data generation and output based on data queue parameters.

28. The apparatus of claim 24 wherein the means for storing includes a first in first out (FIFO) buffer that stores the generated random insertion data as entries therein and wherein the means for filtering incoming data further includes means for comparing the incoming data to at least one entry of stored random insertion data and deleting the at least one entry of stored random data and deleting the incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

29. The apparatus of claim 26 wherein the actual data is keystroke data from a keystroke source, wherein the random insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

30. A storage medium comprising memory containing executable instructions that when read by one or more processing units, causes one or more processing units to:

    generate insertion data for insertion as part of the incoming data, for a software application;

    store the generated insertion data; and

    filter received incoming data containing actual data and the insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data.

31. The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to process the actual data resultant from filtering for use by the software application.

32. The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to:

    receive the generated insertion data and actual data from a data input source; and

    queue the insertion data and the actual data for output as the incoming data.

33. The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to analyze foreground indication data and enable generation of the random insertion data in response to the foreground indication data.

34. The storage medium of claim 33 containing executable instructions that when read by one or more processing units, causes one or more processing

units to control timing of random insertion data generation and output based on data queue parameters.

35. The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to store the generated insertion data as entries in a first in first out (FIFO) buffer and compare the incoming data to at least one entry of stored insertion data and delete the at least one entry of stored data and delete the incoming data when the stored data is the same as the incoming data, and output the incoming data to a data processor when the stored data from the entry is different from the incoming data.

36. The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to store a list of data representing data to be randomized;

randomly select data from the list of data as random insertion data; and

format the random insertion data in a same format as actual data.

37. The storage medium of claim 36 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized key stroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.